УДК 004.75

СВІРІН П.В.

МЕТОДИКА ВДОСКОНАЛЕННЯ БРОКЕРА ДЛЯ NORDUGRID ARC

Для забезпечення вимог користувачів з продуктивності та ефективності виконання завдань грід – система повинна реалізувати ефективний алгоритм розподілу завдань між доступними на даний час обчислювальними ресурсами. Основна мета такого балансування навантаження в грід-системі – скоротити загальний час виконання завдання користувача і забезпечить ефективність використання обчислювальних ресурсів.

Мета даної роботи – розробка методики створення розподіленого брокера для ARC 2.0 із застосуванням сучасних можливостей платформи. Застосовуючи цю методику користувач може розробити власний брокер, який буде враховувати специфіку середовища чи певної категорії задач, для ефективного розподілу завдань серед обчислювальних ресурсів і інтегрувати його в платформу ARC.

In order to provide users with performance and task execution effectivity GRID has to implement an effective brokering algorithm. The main goal of such load balancing in GRID is to decrease the overall execution time and make utilization of the computing resources effective.

The purpose of the work is to develop the methodology on distributed broker implementation for Nordugrid ARC 2.0 using its modern features. Using this methodology users can implement his own brokers that will take into account specifics of environment or certain task category in order to distribute tasks among the available resources in an effective way and integrate this broker into ARC platform.

1. Вступ

Незважаючи на те, що алгоритми балансування навантаження обчислювальних ресурсів у грід системі вже досить давно досліджуються і існує багато вже, як готових алгоритмічних рішень, так і програмних реалізацій, інтенсивний розвиток грід – технологій, вдосконалення проміжного програмного забезпечення робить проблему балансування навантаження постійно актуальною і інтерес до дослідження в цій області не зменшується. Основна мета такого балансування навантаження в грід-системі – скоротити загальний час виконання завдання користувача і забезпечить ефективність використання обчислювальних ресурсів.

Грід–інфраструктура України побудована з використанням програмного забезпечення проміжного рівня (middleware) ARC (Advanced Resource Connector), що також відоме під назвою проекту NorduGrid [3].

В ARC, як с версії 0.8 так і в нової версії ARC 2.0 використовується як основний принцип максимально повна децентралізацію, тому на кожному робочому місці користувача Грід-мережі встановлюється персональний брокер, функція якого – вибір найкращого ресурсу для виконання завдання користувача, яке необхідно виконати в Грід-мережі.

В УНГ на даний момент використовується випадковий вибір ресурса, який не враховує поточного стану існуючих ресурсів. Для більш оптимального розподілу навантаження серед ресурсів необхідно розробити власні брокери, які будуть враховувать як поточний стан ресурсів, так і політику за якою відбувається балансування навантаження.

Слід відмітити, що до комплекту Nordugrid ARC входять лише найпростіші політики, отже, запропоновані методики можуть використовуватись не тільки в УНГ, а й для інших сегментів і віртуальних, організацій зі специфічними та загальними типами завдань,

2. Постановка задачі для УНГ

Основними задачами, для яких необхідний Грід, є:

- велика кількість задач з невеликими вимогами щодо ресурси. Такі задачі виконуються протягом короткого часу;
- велика кількість задач з великими вимогами щодо ресурсів і які виконуються протягом довгого часу.

Прикладами таких задач є:

- обробка даних експерименту ALICE. Звичайно така задача вимагає 1 процесор, дані передаються на ресурс прямо при обчисленні, максимальний час виконання – 24 години. Кількість таких задач може доходити до сотень тисяч;
- обчислення задач молекулярної динаміки.

Цей клас задач вимагає велику кількість процесорів, подає на обчислення невелику кількість даних, максимальний час розрахунку таких задач – місяці. Кількість таких задач може доходити до тисяч.

Таким чином, використання єдиної стратегії для розподілу різних класів задач не є ефективним. Варіантом рішення цієї проблеми стали спеціалізовані грід системи, такі як AliEN Grid, WeNMR. Однак, кількість класів задач є дуже великою і неможливо розробити систему для кожного класу задач.

Особливосстями украинской Грід інфраструктури є:

- 38 кластеров, малих за своїми обчислювальними потужностями [13];
- в наявності є тільки 2 ресурси з великою обчислювальною потужністю;
- всі ресурси працюють під керуванням ARC;
- різноманітна тематика розрахунків: молекулярна динамика, фізика, хімія, астрономія і т.і., велика кількість віртуальних організацій.

Особливостями брокеру ресурсів в Nordugrid ARC є:

- наявність лише спрощених політик розподілу задач
- орієнтація роботи системи на обробку результатів експеримента ATLAS, в якому переважають короткі задачі з невеликими об'ємами даних. Спеціально для цього експеримента було розроблено брокер, що робить висновок щодо цільового ресурса враховуючи об'єм необхідних даних в кеші обчислювального ресурса і, таким чином, зменшуючи час передачі даних.

Таким чином, в українському сегменті немає брокерів, що підходять для оптимального розподілу задач всіх класів.

В реальності задачі, що вимагають 10-30 процесорів направляються на виконання на кластер інститута кібернетики і очікують в черзі протягом днів. Туди ж можуть потрапляти більш короткі задачі і теж простоювать в черзі.

Отже, мета оптимального брокера для УНГ:

- направлення коротких задач – на більш слабкі ресурси
- направлення довгих задач – на більш потужні.

3. Методика побудови брокера ресурсів

3.1. Загальні положення

Загальна послідовність кроків при подачі завдання полягає в:

1. Опиті інформаційної системи;
2. Відборі кандидатів, характеристики яких відповідають вимогам завдання;
3. Отримання додаткової інформації від додаткових джерел інформації про ресурси;
4. Ранжування списку кандидатів згідно алгоритму брокера:
5. Подача завдання на ресурс.

Для реалізації алгоритму власного брокера необхідно реалізувати кроки 2-4.

Для побудови ефективного брокера необхідно:

- визначити сферу завдань і вимоги середовища, для яких брокер може ефективно робити розподіл між ресурсами;
- забезпечити отримання інформації від інформаційної системи;
- визначити критерії відкидання кандидатури ресурса;
- визначити набір характеристик, за якими буде проводитись оцінка завантаженості обчислювального кластеру;
- розробити порядок кроків для ранжування списку кластерів та відсіювання тих кластерів, що не підходять під вимоги завдання, для якого ведеться пошук кластеру призначення.
- розробити процедуру визначення черговості подачі завдань при необхідності розподілити декілька завдань одночасно.

Існують два методи ранжування ресурсів:

- послідовне оцінювання і відкидання ресурсів за кожним критерієм
- розрахунок інтегральної характеристики для кожного ресурса і подальше ранжування за цією характеристику.

Другий підхід вже розглядався в декількох статтях, зокрема [2], в якій автори будували інтегральну оцінку ресурса за його питомою продуктивністю, об'ємом пам'яті, кількістю процесорів, завантаження процесорів протягом попередньої хвилини та коефіцієнтом мережевого завантаження для ресурса.

Платформа Nordugrid ARC 2.0 надає можливість створення веб-сервісів для реалізації служб спеціального призначення. Отже, можливо реалізувати сервіс, що буде повертати розширену інформацію про ресурс. Ще одним джерелом розширеної інформації можуть бути

системи моніторингу, такі як Nagios та ін. В статті [1] було проведено огляд варіанту побудови брокера для Українського сегмента грід [4]. Розглянутий варіант брокера не вимагав встановлення сервісів, що знаходяться поза межами платформи nordugrid ARC.

З архітектурної точки зору брокери можна поділити на:

- постійні – модулі скомпільовані в програмі
- змінні – такі модулі можливо змінювати без перекомпіляції повного коду клієнтського ПЗ. Наприклад, можливо періодично звантажувати оновлення для брокера в рамках деякого проекту. Код алгоритму в даному випадку написано скриптовою мовою Lua. Під час запуску програми подачі завдання даному скрипту передається список ресурсів-кандидантів, на виході скрипт повертає ранжований список ресурсів.

З алгоритмічної точки зору брокери можна поділити на:

- такі, що обчислюють загальної оцінки для ресурса. При цьому можуть враховуватись як статичні і динамічні характеристики ресурса, так і статичні вимоги завдання до ресурсів, що дозволяє проводити балансування навантаження серед ресурсів згідно певної політики;
- такі, що використовують табличний метод – ранжування ресурсів проводиться згідно з вже накопиченими відомостями щодо довжини виконання певних видів завдань, що існують у проекті.

Основні джерела інформації про ресурси Nordugrid ARC:

- AREX – сервіс, що контролює виконання завдань. Повертає список статичних та динамічних характеристик, що описуються схемою GLUE 2;
- Infosys – інформаційна система Nordugrid ARC для версій 0.8 і нижче. Побудована на основі OpenLDAP і вміщує в собі список зареєстрованих обчислювальних ресурсів та їх поточний стан для статичних та динамічних характеристик;
- ISIS – являє собою реєстр сервісів, з яких можна отримати інформацію про ресурси.

Додаткові можливі джерела:

- додаткові сервіси платформи Nordugrid Arc.
- системи моніторингу Network Weather Service, Ganglia та інші;
- інші джерела.

Наприклад, з поширеною в ГРІД-середовищах системи Використання даних з Ganglia:

- кількість працюючих вузлів ресурса на поточний момент
- мережевий стан ресурса
- завантаження кластера протягом останнього часу

3.2. Характеристики для ранжування ресурсів

3.2.1. Статичні характеристики

Статичні характеристики ресурса отримуються шляхом прямого опитування сервіса AREX або інформаційної системи Infosys. Характеристики, отримані таким шляхом використовуються на першому кроці брокера для відкидання ресурсів, які гарантовано не відповідають вимогам завдання.

Статичні характеристики відображають характеристики обчислювальних ресурсів, що не змінюються з часом. Більшість таких характеристики знаходяться в описі ресурса в інформаційній системі. Прикладами статичних характеристик можуть бути:

- кількість процесорів на ресурсі
- кількість вузлів на ресурсі;
- об’єм пам’яті обчислювального
- продуктивність обчислювального ресурса

3.2.2. Динамічні характеристики

Динамічні характеристики є такими характеристиками, які змінюються з часом і брокер веде ранжування обчислювальних ресурсів, враховуючи значення таких характеристик на момент подачі завдання. Базові динамічні характеристики ресурса (такі як довжина черги) можливо отримати з інформаційної системи і розширити їх список використанням сервісів Nordugrid ARC.

Для отримання розширеної інформації щодо статичних характеристик ресурса та динамічних характеристик можливо використовувати додаткові джерела інформації. Частина динамічної інформації щодо стану ресурса зберігається в інформаційній системі, але недоліком такого підходу є те, що на момент вибору ресурса вона може бути застарілою.

Прикладом використання додаткових сервісів серед брокерів Nordugrid ARC є брокер Data, який опитує сервіс, що повертає об'єм даних, що необхідні для виконання завдання, в кеші ресурса. Недоліком даного підходу є те, що в разі відсутності такого сервісу на ресурсі-кандидаті цей ресурс відкидається. Отже, важливою задачею є розробка механізму компенсації відсутності запису про ресурс в додаткових джерелах інформації і не відкидати їх.

Приклади динамічних характеристик:

- кількість задач в черзі на обчислювальний ресурс;
- завантаженість ресурса;
- доступний дисковий об'єм.

3.2.3. Характеристики, специфічні для задачі

У випадку, в якому необхідно розподілювати серед обчислювальних ресурсів завдання, що пов'язані між собою певним чином, брокер може використовувати додаткові характеристики, що є специфічними для даної групи завдань [10][11] Наприклад, в екпериментах ATLAS [5] та ALICE [6] важливим для швидкості виконання завдань є наявність вже оброблених даних у кеші обчислювального ресурса. Розширені параметри завдання, що відсутні в мовах опису xRSL та JSDL [8][9], можливо вказати в централізованій базі даних завдання (наприклад, в ATLAS це prodDB, що базується на СУБД Oracle [5]). Процес виконання може контролюватись сервісом супервізора, напр. Eowyn [7] чи GANGA[12].

При наявності в рамках певного проекту набору однотипних завдань можливо також використовувати базу даних відповідності типу завдання та конфігурації ресурса до необхідного часу для виконання завдання.

3.3 Специфіка реалізації брокера в Nordugrid ARC 2.0.

В Nordugrid ARC 2.0 модуль брокера реалізується в якості плагіна, що завантажується програмою arcsub. Кожен такий плагін реалізовує специфічний для брокера метод відкидання ресурса, якщо його характеристики не задовольняють завданню, а також метод ранжування ресурсів, який визначає більш підходящим один з двох ресурсів, що подаються на вхід. Даний метод використовується в глобальній процедурі сортування ресурсів.

4. Приклади реалізації алгоритма брокера ресурсів

4.1. Побудова інтегральної характеристики із застосуванням виключно інформації з веб-сервісу платформи ARC.

В статті [14] описаний алгоритм побудови інтегральної оцінки ресурса із застосуванням додаткових сервісів платформи ARC. Використовується сервіс, що повертає повну інформацію щодо черг на ресурсі. При відсутності необхідної інформації щодо черг застосовується методика компенсація відсутності цієї інформації.

Розраховуються загальні довжина черги для кожного ресурса

$$N_i = \sum_{j=1}^{n}(L_j + L_{ja}) \qquad (1)$$

, де N_i – загальна довжина черги на і-му ресурсі, n – кількість черг, що існують на LRMS і-го ресурса, L_i- довжина j-ї черги на даному ресурсі, L_{ja} – довжина черги на рівні Nordugrid ARC, тобто кількість задач, що лише очікують на подачу на рівень LRMS.

Для кожного ресурса генерується коефіцієнт, що розраховується наступним чином:

$$M = rand(0, \frac{1}{2^S}) \qquad (2)$$

де $S=N_i$, якщо можливо отримати повну інформацію про черги на ресурсі з додаткового сервіса, або $S=N_i+1$, якщо такого сервіса на ресурсі немає.

Проводиться ранжування ресурсів за спаданням коефіцієнту M.

4.2. Побудова інтегальної характеристики із застосуванням інформації з Ganglia та веб-сервісу платформи ARC.

Коефіцієнт розраховується схожим чином, але використовуються додаткові дані з різних інформаційних джерел.

Приклад формули розрахунку коефіцієнта:

$$K = K_{nodes\_up}K_{cache}K_q K_{network\_load}K_{mem} \qquad (3)$$

де K_{nodes_up} – кількість активних вузлів на обчислювальному ресурсі, K_{cache} – коефіцієнт наявності необхідних для виконання завдання файлів в кеші даного обчислювального ресурса, K_q – коефіцієнт заповненості черг обчислювального ресурса, $K_{network\_load}$ – коефіцієнт завантаженості мережі на обчислювальному ресурсі, K_{mem} – коефіцієнт завантаженості пам'яті даного ресурса. Коефіцієнти K_{nodes_up},

$K_{network\_load}$, K_{mem} отримуються з системи моніторінгу Ganglia, коефіцієнти K_{cache}, K_q отримуються опитуванням веб-сервіса для платформи ARC. При відсутності відомостей в джерелах інформації щодо даного ресурса проводиться компенсація, подібна до описаної вище.

Описані методи з компенсацією недостатньої кількості додаткових інформаційних ресурсів дозволяють брати участь у ранжуванні ресурсам, на яких не встановлені додаткові інформаційні сервіси, або вони не зареєстровані в системах моніторинга. Така ситуація може виникнути в разі експериментального використання того чи іншого джерела, коли не ще немає чіткого рішення щого його застосування.

4.3. Ранжування згідно табличних даних щодо довжини виконання завдання.

Даний метод підходить для віртуальних організацій, які проводять обробку даних в рамках одного проекту, в якому завдання можна розділити на класи і обчислити залежність середньої довжини виконання завдання від об'єму даних, які необхідно обробити та типу системи на цільовому обчислювальному ресурсі. Якщо завдання неможливо віднести до будь-якого з відомих типів, то йому призначається середня довжина виконання зі всіх зареєстрованих типів завдань.

Прикладом розподілу завдань на класи може виступать певний формат опису завдання, що затверджено для проекту.

В даному варіанті застосовуються наступні типи інформаційних ресурсів:

1) інформаційна система Nordugrid ARC, що показує кількість і типи завдань, які ще не надійшли на PBS;
2) сервіс з прикладу 1, що показує кількість завдань на інших чергах обчислювального ресурса;
3) база даних, в якій зберігаються типи завдань та середні довжини їх виконання.

Розраховується оціночна довжина виконання завдань на ресурсі кандидаті:

$$L = \sum_{i=1}^{n} l_i \qquad (4)$$

де l_i – довжина виконання і-го завдання на даному ресурсі.

Ресурси ранжуються за зростанням оціночної довжини виконання завдань. Таким чином, на завдання надійде на ресурс з мінімальною довжиною виконання завдань, які надійшли раніше.

Також можливо відслідковувати фази виконання завдань, що дозволить більш точно розраховувать оціночну довжину виконання завдань.

До переваг даного метода можна віднести можливість визначення часу початку виконання завдання, що подається на ресурс. До недоліків – необхідність збору і обробки статистики виконання завдань.

4.4. Змінний брокер, що реалізується окремим скриптом.

Скрипт брокера періодично оновлюється з центрального сервера. Брокер ARC передає через інтерфейс Lua чи Python для мови C++ список ресурсів кандидатів, який ранжується всередині скрипта і повертається назад для подачі завдання. Таким чином можливо реалізувати будь-який алгоритм брокера і оновлювати його залежно від вимог віртуальної організації, в якій він використовується, без перекомпіляції клієнтського інструментарію.

Даний підхід відноситься до категорії архітектурних і вимагає лише інтерфейсів для таких скриптових мов на стороні користувача Грід та системи оновлення скриптів брокера.

Заключення

В статті проведено огляд принципів побудови брокерів для Nordugrid ARC, а також огляд можливих джерел інформації для ранжування списку ресурсів брокером. Розглянуто методику формування такої інтегральної оцінки за допомоги інформаційних джерел з комплекту Nordugrid ARC, а також комбінацію інформації з цих джерел і з даними зі сторонніх джерел, які надають додаткову інформацію про обчислювальні ресурси. Представлено варіанти реалізації алгоритму розподілу навантаження.

Отримані результати дозволяють проводити вибір джерел, що роблять можливим адекватних оцінити поточний стан ресурса та вибирати найкращий алгоритм брокера або його архітектури згідно умов обчислювального середовища чи віртуальної організації. Досліджено метод ранжування обчислювальних ресурсів, що мають неповну інформацію про себе в додаткових джерелах.

Використовуючи запропоновані методики можливо підібрати максимально відповідний

метод ефективного розподілу завдання між обчислювальними ресурсами згідно вимог до продуктивності обчислень і завантаження ресурсів. Також за даними методикою також можливо будувати інші варіанти реалізації із застосуванням інших додаткових джерел інформації.

Список літератури

1. А. Петренко Алгоритм оцінки завантаженості ГРІД-сайту / Петренко А.І., Свістунов С.Я. Свірін П.В // Системный анализ и информационные технологии : 13-я международная научно-техническая конференция «САИТ-2011», 23-28 мая 2011, Киев, Украина : материалы. – К. : УНК "ИПСА" НТУУ "КПИ", 2011. – С. 388.
2. Y. Chao-Tung. A Grid Resource Broker with Network Bandwidth-Aware Job Scheduling for Computational Grids / Y. Chao-Tung, C. Sung-Yi, C. Tsui-Ting. // Advances in Grid and Pervasive Computing. – 2007 – Vol. 4459. – pp.1 – 12.
3. Веб-сайт Nordugrid ARC. [Електронний ресурс] – Режим доступу: http://www.nordugrid.org
4. А.Загородний. Украинский академический грид: Українсько-македонський науковий збірник, Випуск 4 / А.Загородний, Г. Зиновьев, Е. Мартынов, С. Свистунов. – Київ 2009, Вид.Національна бібліотека України імені В.І.Вернадського. – с.140-150.
5. A. Read. Complete Distributed Computing Environment for a HEP Experiment: Experience with ARC-Connected Infrastructure for ATLAS. [Електронний ресурс] / A. Read, A. Taga, F. Ould-Saada, K. Pajchel, B. H. Samset, D. Cameron. – Режим доступа: http://www.nordugrid.org/documents/chep07-atlas.pdf
6. B. Beckles. Report on Swegrid/NorduGrid and the relationship between Swegrid and EGEE. [Електронний ресурс] – Режим доступа: https://www.jiscmail.ac.uk/cgi-bin/ webadmin?A3=ind0405&L=ETF&E=BASE64&P=361112&B=---559023410-1144747756-1085753675%3D%3A6504&T=APPLICATION%2Fpdf;%20name=%22Swegrid_report.pdf%22&N=Swegrid_report.pdf
7. J. Kennedy. ATLAS Production System. [Електронний ресурс] – Режим доступа: http://www.etp.physik.uni-muenchen.de/dokumente/talks/jkennedy_dpg07.pdf
8. Extended Resource Specification Language. [Електронний ресурс] – Режим доступа: http://www.nordugrid.org/documents/xrsl.pdf
9. O. Smirnova. NorduGrid's Extended Globus RSL (xRSL) vs JSDL, a comparison. [Електронний ресурс] / O. Smirnova, M. Niinimaki. – Режим доступа: http://www.nordugrid.org/documents/jsdl-vs-xrsl-revised2.pdf
10. J. C. Werner. Grid computing in High Energy Physics using LCG: the BaBar experience. [Електронний ресурс] – Режим доступа: http://www.gridpp.ac.uk/papers/ahm06_werner.pdf
11. L. Boyanov. On the employment of LCG GRID middleware. [Електронний ресурс] / L. Boyanov, P. Nenkova. – Режим доступа: http://ecet.ecs.ru.acad.bg/cst05/Docs/cp/SII/II.11.pdf
12. F. Brochu. Ganga: a tool for computational-task management and easy access to Grid resources [Електронний ресурс] / F. Brochu, U. Egede, J. Elmsheuser et al.. – Режим доступа: http://ganga.web.cern.ch/ganga/documents/pdf/ganga_cpc09.pdf
13. Grid Monitor. [Електронний ресурс] – Режим доступа: http://gridmon.bitp.kiev.ua/
14. А. Петренко. Гібридний Алгоритм брокера для Nordugrid ARC 2.0. / Петренко А.І, Свистунов С.Я., Свірін П.В. // HPC UA 2012: друга міжнародна конференція «Високопродуктивні обчислення», 8-10 жовтня, 2012, Київ, Україна 8-10 жовтня : матеріали. – К. : Національна академія наук України, 2012. – с. 275.